CLAIMS:

1. (Currently Amended): An architected register file system at least configured to utilize a plurality of threads, comprising:

a plurality of register files, wherein each register file of the plurality of register files at least corresponds to ~~at least one~~ a respective thread of the plurality of threads;

a plurality of Status and Control Registers (SCR), wherein each SCR corresponds to ~~at least one register file of the plurality of register files~~ a respective thread of the plurality of threads; and

a plurality of control bit sets, wherein each control bit set corresponds to at least one SCR, and wherein each control bit set is at least configured to allow a thread associated with an associated SCR to utilize other register files associated with other threads.

2. (Original): The architected register file system of claim 1, wherein the architected register file system further comprises a decoder, wherein the decoder at least determines desired operations for an instruction.

3. (Original): The architected register file system of claim 1, wherein plurality of control bits further comprise a plurality of bit doublets, wherein a first bit of a bit doublet corresponds to a read function, and wherein a second bit of the bit doublet corresponds to a write function.

4. (Original): The architected register file system of claim 3, wherein the architected register file system further comprises:

an address control, wherein the address control at least determines addresses with the plurality of register files; and

at least one execution unit, wherein the execution is at least configured to perform the operations of a input instruction within the plurality of register files.

5. (Currently Amended): The architected register file system of claim 3, wherein the plurality of bit doublets further comprises that each bit ~~doubled~~ doublet at least corresponds to enabling the use of at least one register file associated with another thread.

6. (Original): The architected register file system of claim 5, wherein each bit doublet of the plurality of bit doublets further comprises:

at least one bit is at least configured to correspond to a read function, wherein a logic high or `1` enables the first thread to read from another register file; and

at least one bit is at least configure to correspond to a write function, wherein a logic high or `1` enables the first thread to write to another register.

7. (Currently Amended): A method for utilizing a plurality of register files with associated SCRs in a multithread system, wherein each register file is at least associated with one thread of a plurality of threads, comprising:

receiving an instruction for a first thread of the plurality of threads, wherein the first thread is at least associated with a first SCR;

decoding the instruction to at least determine performance operations;

determining if the first thread is enabled to at least utilize register files associated with other threads; and

executing the instruction, wherein the step of executing ~~at least~~ utilizes ~~whatever register files that are enabled~~ at least one register file associated with a second thread of the plurality of threads.

8. (Currently Amended): The method of claim 7, wherein the step of determining if the first thread is enabled, further comprises measuring logical levels of control bits associated with the first SCR, wherein the control bits comprise a plurality of bit doublets, and wherein each bit ~~doubled~~ doublet at least corresponds to enabling the use of at least one register file associated with another thread.

9. (Original): The method of claim 8, wherein the step of measuring further comprises determining if any bits are `1` or logic high, wherein the `1` or the logic high enables the

first thread to read or write to another register file.

10-15. (Canceled)

16. (New): A method for utilizing a plurality of register files in a multithread system, the method comprising:

receiving an instruction for a first thread having an operations code field, a write field, and one or more read fields, wherein the operations code field defines a desired operation for the instruction, wherein the write field defines an address location to which a result of the operation is to be stored, and wherein the at least one read field defines an address location from which data is to be read for the operation;

decoding the instruction;

setting a first status and control register associated with the first thread and a second status and control register associated with a second thread based on the decoding of the instruction, wherein a first register file is associated with the first thread and a second register file is associated with the second thread;

determining whether the first thread is permitted to utilize the second register file associated with the second thread based on at least one of the first status and control register or the second status and control register; and

if the first thread is permitted to utilize the second register file, performing the operation utilizing the second register file by writing to or reading from the second register file associated with the second thread.

17. (New): The method of claim 16, wherein performing the operation comprises:

reading data from the second register file based on an address in a read field within the one or more read fields.

18. (New): The method of claim 16, wherein performing the operation comprises:

writing a result of the operation to the second register file based on an address in the write field.

19. (New): An apparatus for utilizing a plurality of register files in a multithread system, the apparatus comprising:

a decoder that receives and decodes an instruction for a first thread having an operations code field, a write field, and one or more read fields, wherein the operations code field defines a desired operation for the instruction, wherein the write field defines an address location to which a result of the operation is to be stored, and wherein the at least one read field defines an address location from which data is to be read for the operation;

a first status and control register associated with the first thread;

a first register file associated with the first thread;

a second status and control register associated with a second thread;

a second register file associated with the second thread, wherein the decoder sets the first status and control register and the second status and control register based on the decoding of the instruction, wherein a first register file is associated with the first thread and a second register file is associated with the second thread;

an address control that enables or disables access to the first register file and the second register file based on the first status and control register and the second status and control register, wherein the address control enables the first thread to access the second register file if the first thread is permitted to utilize the second register file; and

one or more execution units that perform the operation utilizing the second register file by writing to or reading from the second register file associated with the second thread based on the address control.

20. (New): The apparatus of claim 19, wherein the first status and control register comprises a first control field that indicates whether to enable reading from the second register file.

21. (New): The apparatus of claim 20, wherein the first status and control register comprises a second control field that indicates whether to enable writing to the second register file.